5

APPARATUS AND METHOD TO FACILITATE

SELF-CORRECTING MEMORY

Inventors: James E. Kocol, Ashley N. Saulsbury, and Sandra C. Lee

15

BACKGROUND

Field of the Invention

20   The present invention relates to computer memory. More specifically, the present invention relates to self-correcting memory in a shared memory multiprocessor system.

Related Art

25   Modern computing systems designers are under constant pressure to increase the speed and density of the integrated circuit devices, including the memory devices, within these systems.

        Increasing the density of memory devices, however, can cause the occurrence of "soft errors" to increase, which can lead to erroneous computational

30   results. These soft errors occur at random and are attributable to uncontrollable

1

causes, such as alpha particle radiation. Increased density leads to smaller memory cells within the memory devices and, in turn, smaller charge levels within the cell to indicate the logic state of the cell. The smaller charge levels make a cell more susceptible to soft errors.

5   In an attempt to reduce the impact of soft errors within a memory system, designers have routinely used self-correcting memory systems as described in U.S. Patent number 4,319,356 issued to James E. Kocol and David B. Schuck. These self-correcting memory systems use additional bits within the device for storing an error correcting code, and use error correcting circuitry to correct any cells that

10   have a soft error. In operation, the memory system periodically visits each cell within the memory system and corrects any errors detected in the cell's data. This process is termed "scrubbing the memory."

          There are several methods that can be used to form the error correcting code, and to correct soft errors. In general, the number of bits assigned to the

15   error correcting code determines how many errors the error correcting systems can correct. Commonly available systems include single bit error correction/double bit error detection, and double bit error correction.

          While effective, these error correcting memory systems do not provide error correction on cache memory within multiprocessor shared memory systems.

20   Typically, devices and subsystems such as a central processing unit or an input/output device within these multiprocessor shared memory systems have an associated cache for storing data while it is in use by the device or subsystem. As the system operates, data from the memory system is "checked out" to the cache. While the data is checked out to a cache, correcting errors in the cells in main

25   memory will not correct errors in the cache. If the data is checked out for a long time, it is possible for multiple soft errors to accumulate within the data cell such

2

that the number of errors is beyond the capabilities of the self-correcting memory system.

What is needed is a method and apparatus for eliminating soft errors in data checked out to a cache.

5

SUMMARY

One embodiment of the present invention provides a system that facilitates self-correcting memory in a shared-memory system. This system includes a main memory comprised of dynamic random access memory. A memory controller is

10 coupled to the main memory for reading and writing memory locations and for marking memory locations that have been checked out to a cache. The system also includes a processor cache for storing data currently in use by a central processing unit. A communication channel is coupled to the processor cache and to the memory controller to facilitate communication between these units. The

15 memory controller includes an error detection and correction mechanism, which uses an available error detection and correction system. The memory controller also includes a reading mechanism that is configured to read a data from the processor cache when a currently valid copy of the data is checked out to the processor cache. When the data is returned to the memory subsystem from the

20 cache, the error detection and correction mechanism corrects errors in the data and stores a corrected copy of the data in the main memory.

In one embodiment of the present invention, the error detection and correction mechanism performs single bit error correction/double bit error detection.

25 In one embodiment of the present invention, the error detection and correction mechanism performs double bit error correction.

3

In one embodiment of the present invention, the system includes an input/output cache associated with an input/output device. The reading mechanism is further configured to read the data from the input/output cache when the currently valid copy is checked out to the input/output cache. When the data is returned to the memory subsystem from the input/output cache, the error detection and correction mechanism corrects errors in the data and stores the corrected copy of the data in the main memory.

In one embodiment of the present invention, the system includes a second processor cache. In this embodiment, the reading mechanism is further configured to read the data from the second processor cache when the currently valid copy is checked out to the second processor cache. When the data is returned to the memory subsystem from the second processor cache the error detection and correction mechanism corrects errors in the data and stores the corrected copy of the data in the main memory.

In one embodiment of the present invention, the system includes a marking mechanism within the memory controller. The marking mechanism is configured to mark a location in the main memory to indicate that the data from the location is checked-out to a cache. The cache can be any cache coupled to the system including the processor cache, the input/output cache, or the second processor cache.

In one embodiment of the present invention, the system includes a scrubbing mechanism within the memory controller that is configured to access each location within the main memory periodically. This scrubbing mechanism works in conjunction with the error detection and correction mechanism to detect and correct errors.

In one embodiment of the present invention, the system includes a detecting mechanism coupled to the scrubbing mechanism. The detecting

4

mechanism is configured to detect that a location in the main memory is marked that the data from the location is checked-out to the cache. The reading mechanism is further configured to request a read from the communication channel if the location is so marked.

5   In one embodiment of the present invention, the communication channel is a coherent network.

BRIEF DESCRIPTION OF THE FIGURES

FIG. 1 illustrates computing device 100 in accordance with an

10   embodiment of the present invention.

FIG. 2 illustrates memory controller 104 in accordance with an embodiment of the present invention.

FIG. 3 is a flowchart illustrating the process of correcting memory errors in accordance with an embodiment of the present invention.

15

DETAILED DESCRIPTION

The following description is presented to enable any person skilled in the art to make and use the invention, and is provided in the context of a particular application and its requirements. Various modifications to the disclosed

20   embodiments will be readily apparent to those skilled in the art, and the general principles defined herein may be applied to other embodiments and applications without departing from the spirit and scope of the present invention. Thus, the present invention is not intended to be limited to the embodiments shown, but is to be accorded the widest scope consistent with the principles and features

25   disclosed herein.

<u>Computing Device</u>

5

FIG. 1 illustrates computing device 100 in accordance with an embodiment of the present invention. Computing device 100 can generally include any type of computer system, including, but not limited to, a computer system based on a microprocessor, a mainframe computer, a digital signal processor, a portable computing device, a personal organizer, a device controller, and a computational engine within an appliance. Computing device 100 includes main memory 102, memory controller 104, coherent network 106, cache controller 108, processor cache 110, central processing unit 112, input/output controller 114, input/output cache 116, and input/output device 118.

Main memory 102 stores data associated with computer applications being executed by computing device 100. Memory controller 104 controls main memory 102 and interfaces main memory 102 with coherent network 106. Details of the operation of memory controller 104 are described below in conjunction with the description of FIG. 2.

Coherent network 106 couples various devices and subsystems within computing device 100. In operation, coherent network 106 transports data between the various devices and subsystems and includes signals related to maintaining coherency among the several caches within computing device 100. Details of the operation of coherent network 106 are well known in the art and are not described herein.

Processor cache 110 stores data for central processing unit 112. Typically, the data stored within processor cache 110 is recently accessed data and data stored near the recently accessed data within main memory 102. This allows central processing unit 112 to access data directly from processor cache 110 rather than across coherent network 106 for most access cycles. By accessing data from processor cache 110, central processing unit 112 avoids delays associated with

6

contention on coherent network 106, and increased access time for data stored in main memory 102.

Central processing unit 112 can generally include any type of processor, including, but not limited to, a microprocessor, a mainframe computer, a digital signal processor, a personal organizer, a device controller and a computational engine within an appliance. Central processing unit 112 provides computational and decision making functions for computing device 100.

A practitioner skilled in the art will readily appreciate that cache controller 108, processor cache 110 and central processing unit 112 are duplicated multiple times within multiprocessor systems. Memory controller 104, coherent network 106, cache controller 108, input/output controller 114, and other controllers coupled to coherent network 106 function in concert to ensure data coherency within main memory 102, processor cache 110, input/output cache 116, and any additional cache associated with coherent network 106.

Input/output controller 114 controls input/output cache 116 and input/output device 118. In addition, input/output controller 114 couples input/output cache 116 and input/output device 118 to coherent network 106.

Input/output cache 116 buffers data for input/output device 118 and functions in much the same manner as processor cache 110 described above.

Input/output device 118 is a data interface between devices coupled to coherent network 106 and external devices such as disk drives, tape drives, modems, and the like. Input/output controller 114, input/output cache 116 and input/output device 118 may also be replicated as needed within computing device 100 as will be obvious to a practitioner skilled in the art.

7

Memory Controller

FIG. 2 illustrates memory controller 104 in accordance with an embodiment of the present invention. Memory controller 104 includes error detection and correction mechanism 202, reading mechanism 204, writing

5   mechanism 206, marking mechanism 208, scrubbing mechanism 210 and detecting mechanism 212.

Error detection and correction mechanism 202 functions to correct bit errors in memory locations. Error detection and correction mechanism 202 can be any available error detection and correction mechanism. Typical error detection

10   and correction mechanisms include single bit error correction/double bit error detection mechanisms, and double bit error correction mechanisms. By correcting "soft" bit errors as they occur, data within computing device 100 will be less likely to accumulate errors that are beyond the capability of error detection and correction mechanism 202 to correct these errors.

15   Scrubbing mechanism 210 works in conjunction with error detection and correction mechanism 202. Scrubbing mechanism 210 periodically visits each location in main memory 102 and provides the data to error detection and correction mechanism 202. After any errors are corrected by error detection and correction mechanism 202, the location within main memory 102 is rewritten with

20   the corrected data.

Marking mechanism 208 marks a location within main memory 102 as invalid when the current copy of data from the location has been checked out to a cache on coherent network 106. When the current copy of data is returned to its location within main memory 102, marking mechanism 208 marks the location as

25   valid.

Detecting mechanism 212 works in conjunction with scrubbing mechanism 210. When scrubbing mechanism 210 visits a location that is marked

8

as invalid within main memory 102, detecting mechanism 212 detects that the location has been marked as invalid. When the location is marked as invalid, the system causes the data to be read from the current cache location, corrected if necessary, and rewritten within main memory 102.

5   Reading mechanism 204 requests data from coherent network 106 when detecting mechanism 212 detects that a location within main memory 102 has been marked as invalid. Reading mechanism 204 provides the data returned from coherent network 106 to error detection and correction mechanism 202 so that any errors can be corrected.

10   Writing mechanism 206 writes the corrected data back to main memory 102.

Memory Corrections

FIG. 3 is a flowchart illustrating the process of correcting memory errors
15   in accordance with an embodiment of the present invention. The system starts when scrubbing mechanism 210 determines that it is time to scrub a location within main memory 102 (step 302). Typically, scrubbing mechanism 210 cycles through all memory locations within main memory 102 at a predetermined rate.

If it is time to scrub the location, memory controller 104 accesses the data
20   from the memory location (step 304). Next, error detection and correction mechanism 202 determines if there is an error in the data at the location accessed (step 306). If there is an error in the data, error detection and correction mechanism 202 corrects the error (step 308).

After correcting the error at 308 or if there is no error at 306, detecting
25   mechanism 210 determines if marking mechanism 208 has marked the location as invalid indicating that the data from the location has been checked out to a cache (step 310).

9

If the location has been marked as invalid by marking mechanism 208, reading mechanism 204 reads the data from the current cache location coupled to coherent network 106 and, if necessary, corrects the data from the cache (step 312). After any error in the cache data has been corrected, writing mechanism 5 206 optionally stores the corrected data in the location within main memory 102 (step 314).

The process continues from step 302 so that all locations within main memory 102 can be corrected.

The foregoing descriptions of embodiments of the present invention have 10 been presented for purposes of illustration and description only. They are not intended to be exhaustive or to limit the present invention to the forms disclosed. Accordingly, many modifications and variations will be apparent to practitioners skilled in the art. Additionally, the above disclosure is not intended to limit the present invention. The scope of the present invention is defined by the appended 15 claims.